1. Leer la base y la altura de un triángulo recto, calcular la hipotenusa. Para el cálculo de la raíz cuadrada puede utilizar la serie de Taylor de función exponencial y del logaritmo natural (no agregue esas funciones al lenguaje, la idea es probar ciclos y expresiones aritméticas. Su programa deberá preguntar por la lectura de una siguiente función 1:sí, 2:salir.

\*\*Newton Rapson

**def hipotenusa(): base = base \* base altura = altura \* altura hipo = base + altura raiz = hipo/2 aux = 0 while aux != raiz: aux = raiz raiz = (hipo/aux+aux)/2 end escribir : TuResultadoEs escribir : raiz return var(): base = 0 altura = 0 hipo = 0 raiz = 0 sel = 0 aux = 0 return begin: while sel < 2: escribir : CalcularP1 escribir : SalirP2 sel = input(OPCION) if sel === 1: base = input(CualEsLaBase) altura = input(CualEsLaAltura) call hipotenusa end end end**

def hipotenusa():

base = base \* base

altura = altura \* altura

hipo = base + altura

raiz = hipo/2

aux = 0

while aux != raiz:

aux = raiz

raiz = (hipo/aux+aux)/2

end

escribir : TuResultadoEs

escribir : raiz

return

var():

base = 0

altura = 0

hipo = 0

raiz = 0

sel = 0

aux = 0

return

begin:

while sel < 2:

escribir : CalcularP1

escribir : SalirP2

sel = input(OPCION)

if sel === 1:

base = input(CualEsLaBase)

altura = input(CualEsLaAltura)

call hipotenusa

end

end

end

2. Leer un número entero e imprimir con letras los dígitos que lo forman, por ejemplo: si se lee 8753 imprimir: tres, cinco, siete y ocho. No agregue la operación de residuo, no es necesaria. El número puede tener cualquier número de dígitos (mientras sea un entero).

**def impresora(): if b === 0: escribir : CERO end elif b === 1: escribir : UNO end elif b === 2: escribir : DOS end elif b === 3: escribir : TRES end elif b === 4: escribir : CUATRO end elif b === 5: escribir : CINCO end elif b === 6 : escribir : SEIS end elif b === 7: escribir : SIETE end elif b === 8: escribir : OCHO end elif b === 9: escribir : NUEVE end return var():num=0 a=0 b=0 x = 0 aux1 = 0 aux2 = 0 return begin: num = input(DameUnValor) while num>0 : aux1=num/10 aux1 = aux1\*10 aux2 = num-aux1 num = num/10 b = aux2 escribir : b call impresora end end**

def impresora():

if b === 0:

escribir : CERO

end

elif b === 1:

escribir : UNO

end

elif b === 2:

escribir : DOS

end

elif b === 3:

escribir : TRES

end elif b === 4:

escribir : CUATRO

end

elif b === 5:

escribir : CINCO end

elif b === 6 :

escribir : SEIS

end

elif b === 7:

escribir : SIETE

end

elif b === 8:

escribir : OCHO

end

elif b === 9:

escribir : NUEVE

end

return

var():

num=0

a=0

b=0

x = 0

aux1 = 0

aux2 = 0

return

begin:

num = input(DameUnValor)

while num>0 :

aux1=num/10

aux1 = aux1\*10

aux2 = num-aux1

num = num/10

b = aux2

escribir : b

call impresora

end

end

**3. Calcular el factorial de un número x (que lea) de manera recursiva (como en clase).**

**def modul(): while (i=<x) : f = f\*i i= i+1 end return var(): x = 0 f=1 i = 1 return begin: x = input(numero) call modul escribir : TuResultadoEs escribir : f end**

def modul():

while (i=<x) :

f = f\*i

i= i+1

end

return

var():

x = 0

f=1

i = 1

return

begin:

x = input(numero que quiere factorial)

call modul

print((el valor del factorial de x es):f)

end

Programa 1. 20 puntos.

Diseñe un programa que lea las dimensiones de dos matrices de tamaño máximo de 5x5 (declare las matrices de 5x5).

El programa deberá contar con un menú. El usuario elige la opción de:

a) Multiplicar las dos matrices y dejar el resultado en una tercera matriz, imprimir resultado. Antes deberá verificar que sea posible realizar la multiplicación.

b) Sumar las dos matrices y dejar el resultado en una tercera matriz, imprimir resultado. Antes deberá verificar que sea posible realizar la suma. Utilice métodos para un diseño más claro.

**\*se llena por columnas**

**def sum(): for (i=0: i<rows1:i=i+1): for (j=0: j<cols1:j=j+1): arr3(0,j,i) = arr1(0,j,i) + arr2(0,j,i) end end return def fill(): for (i=0 : i<rows1 : i=i+1): for (j=0 : j<cols1 : j=j+1): arr1(0,j,i) = input(IngresaValor) end end return def fill2(): for (i=0 : i<rows2 : i=i+1): for (j=0 : j<cols2 : j=j+1): arr2(0,j,i) = input(IngresaValor) end end return def mull(): escribir:entre1 if cols1 === rows2 : escribir:entre2 for(i = 0 : i < rows1 : i = i+1): for(j = 0 : j < cols2 : j = j+1): for(k = 0 : k < cols1 : k=k+1): arr3(0,i,j) = arr3(0,i,j) + arr1(0,i,k) \* arr2(0,k,i) end end end end escribir:Sali return var(): rows1 = 5 cols1 = 5 arr1 = mat(0,cols1,rows1) rows2 = 5 cols2 = 5 arr2 = mat(0,cols2,rows2) arr3 = mat(0,cols1,rows2) i=0 j=0 k = 0 sel = 0 return begin: escribir:MullSum escribir: llenar1 call fill escribir: llenar2 call fill2 escribir:arr1 escribir:arr2 while sel < 3: escribir:MulP1 escribir:SUMP2 escribir:SALIRP3 sel = input (Opcion) if sel === 1 : call mull escribir: arr3 end if sel === 2 : call sum escribir: arr3 end end end**

def sum():

for (i=0: i<rows1:i=i+1):

for (j=0: j<cols1:j=j+1):

arr3(0,j,i) = arr1(0,j,i) + arr2(0,j,i)

end

end

return

def fill():

for (i=0 : i<rows1 : i=i+1):

for (j=0 : j<cols1 : j=j+1):

arr1(0,j,i) = input(IngresaValor)

end

end

return

def fill2():

for (i=0 : i<rows2 : i=i+1):

for (j=0 : j<cols2 : j=j+1):

arr2(0,j,i) = input(IngresaValor)

end

end

return

def mull():

escribir:entre1

if cols1 === rows2 :

escribir:entre2

for(i = 0 : i < rows1 : i = i+1):

for(j = 0 : j < cols2 : j = j+1):

for(k = 0 : k < cols1 : k=k+1):

arr3(0,i,j) = arr3(0,i,j) + arr1(0,i,k) \* arr2(0,k,i)

end

end

end

end

escribir:Sali

return

var():

rows1 = 5

cols1 = 5

arr1 = mat(0,cols1,rows1)

rows2 = 5

cols2 = 5

arr2 = mat(0,cols2,rows2)

arr3 = mat(0,cols1,rows2)

i=0

j=0

k = 0

sel = 0

return

begin:

escribir:MullSum

escribir: llenar1

call fill

escribir: llenar2

call fill2

escribir:arr1

escribir:arr2

while sel < 3:

escribir:MulP1

escribir:SUMP2

escribir:SALIRP3

sel = input (Opcion)

if sel === 1 :

call mull

escribir: arr3

end

if sel === 2 :

call sum

escribir: arr3

end

end

end

Programa 2. 20 puntos.

Diseñe un programa que lea la dimensión de un vector de máximo 10 elementos y efectúe un sort ascendente de los elementos. - Imprima el vector ya ordenado. - Pregunte si se desea ordenar otro vector. Utilice métodos para un diseño más claro.

**def sort(): for (i=0: i<rows:i=i+1): for (j=i+1: j<rows:j=j+1): aux1 = arr(0,0,i) aux2 = arr(0,0,j) if aux1>aux2: arr(0,0,i) = aux2 arr(0,0,j) = aux1 end end end return def fill(): while i < rows: arr(0,0,i) = input(ingresarvalor) i = i+ 1 end escribir:DiezTerminos return var(): sel = 0 rows = 10 arr = mat(0,0,rows) aux1 = 0 aux2 = 0 i = 0 x = 0 j = 0 return begin: escribir:CodigoOrdenar while sel < 2: escribir : CalcularSort1 escribir : Salir2 sel = input(OPCION) if sel === 1: escribir: hola call fill i = 0 escribir:arr call sort i = 0 escribir:arr j = 0 end end end**

def sort():

for (i=0: i<rows:i=i+1):

for (j=i+1: j<rows:j=j+1):

aux1 = arr(0,0,i)

aux2 = arr(0,0,j)

if aux1>aux2:

arr(0,0,i) = aux2

arr(0,0,j) = aux1

end

end

end

return def fill():

while i < rows:

arr(0,0,i) = input(ingresarvalor)

i = i+ 1

end

escribir:DiezTerminos

return

var():

sel = 0

rows = 10

arr = mat(0,0,rows)

aux1 = 0

aux2 = 0

i = 0

x = 0

j = 0

return

begin:

escribir:CodigoOrdenar

while sel < 2:

escribir : CalcularSort1

escribir : Salir2

sel = input(OPCION)

if sel === 1:

escribir: hola

call fill

i = 0

escribir:arr

call sort

i = 0

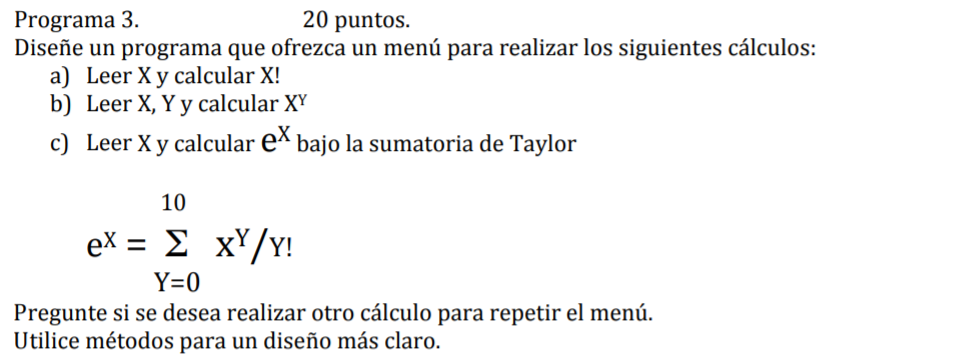
escribir:arr

j = 0

end

end

end



**def exp(): i = 0 e = 1 while (i<y): e = e \* x i=i+1 end return def factorialy(): i = 1 f = 1 while (i=<y) : f = f\*i i= i+1 end return def factorial(): f = 1 i = 1 while (i=<x) : f = f\*i i= i+1 end return def taylor():while (y<10): call exp call factorialy t = t + e/f y = y +1 end return var(): f = 0 sel = 0 x = 0 y = 0 i = 0 t = 0 e=0 return begin: escribir:CodigoTaylor while sel < 4: sel = input(OPCION) if sel === 1: x = input(FactorialDe) escribir: hola call factorial escribir: resultado escribir: f end if sel === 2: x = input(Numero) y = input(Exponente) escribir: hola call exp escribir: Resultado escribir: e end if sel === 3: x = input(exponente) y = 0 f = 0 t = 0 escribir: hola call taylor escribir: Resultado escribir: t end end end**

def exp():

i = 0

e = 1

while (i<y):

e = e \* x

i=i+1

end

return

def factorialy():

i = 1

f = 1

while (i=<y) :

f = f\*i

i= i+1

end

return

def factorial():

f = 1

i = 1

while (i=<x) :

f = f\*i

i= i+1

end

return

def taylor():

while (y<10):

call exp

call factorialy

t = t + e/f

y = y +1

end

return

var():

f = 0

sel = 0

x = 0

y = 0

i = 0

t = 0

e=0

return

begin:

escribir:CodigoTaylor

while sel < 4:

sel = input(OPCION)

if sel === 1:

x = input(FactorialDe)

escribir: hola

call factorial

escribir: resultado

escribir: f

end

if sel === 2:

x = input(Numero)

y = input(Exponente)

escribir: hola

call exp

escribir: Resultado

escribir: e

end

if sel === 3:

x = input(exponente)

y = 0

f = 0

t = 0

escribir: hola

call taylor

escribir: Resultado

escribir: t

end

end

end

Programa 4. 20 puntos. Lea un cubo e imprímalo multiplicado por la constante 2 (multiplicar cada elemento por 2).

**def mull(): for (i=0: i<rows:i=i+1): for (j=0: j<cols:j=j+1): for (z=0: z<deps:z=z+1): arr(z,j,i) = arr(z,j,i) \* 2 end end end return def fill(): for (i=0: i<rows:i=i+1): for (j=0: j<cols:j=j+1): for (z=0: z<deps:z=z+1): arr(z,j,i) = input(IngresaValor) end end end return var(): rows = 3 cols = 3 deps = 3 arr = mat(deps,cols,rows) i=0 j=0 z=0 return begin: escribir:MatCubo call fill escribir: arr call mull escribir: arr end**

def mull():

for (i=0: i<rows:i=i+1):

for (j=0: j<cols:j=j+1):

for (z=0: z<deps:z=z+1):

arr(z,j,i) = arr(z,j,i) \* 2

end

end

end

return

def fill():

for (i=0: i<rows:i=i+1):

for (j=0: j<cols:j=j+1):

for (z=0: z<deps:z=z+1):

arr(z,j,i) = input(IngresaValor)

end

end

end

return

var():

rows = 3

cols = 3

deps = 3

arr = mat(dep,cols,rows)

i=0

j=0

z=0

return

begin:

escribir:MatCubo

call fill

escribir: arr

call mull

escribir: arr

end